

IN THE CLAIMS:

Please amend the claims as follows:

1-42. (Cancelled)

43. (New) A computer-implemented method for debugging code, comprising:

while execution is halted at a first point in the code, receiving a user selection of a target call site of the code, the user selection being made through a graphical user interface, and the target call site comprising a call to any one of a plurality of overriding methods; and

in response to the user selection, setting a run-into breakpoint at each respective entry point of the plurality of overriding methods, wherein the run-into breakpoint is configured to halt subsequent execution only when the target call site calls one of the plurality of overriding methods.

44. (New) The method of claim 43, further comprising:

while the code is under debug:

encountering one of the plurality of run-into breakpoints, during execution of the code under debug; and

determining whether the target call site called the one of the plurality of overriding methods.

45. (New) The method of claim 44, further comprising:

if the target call site called the one of the plurality of overriding methods:
halting execution of the code.

46. (New) The method of claim 43 wherein the code is object-oriented.

47. (New) A computer-implemented method for debugging code, comprising:

while execution is halted at a first point in the code, receiving a user selection of a target call site of the code, the user selection being made through a graphical user

interface, and the target call site comprising a call to any one of a plurality of overriding methods;

in response to receiving the user selection, setting a run-into breakpoint at each respective entry point of the plurality of overriding methods;

determining call context information identifying a location of the target call site in the code;

encountering one of the run-into breakpoints;

determining whether the target call site called one of the plurality of overriding methods, based on the call context information; and

halting execution of the code at the target call site.

48. (New) The method of claim 47 where the call context information unambiguously identifies the code.

49. (New) The method of claim 47, wherein determining whether the target call site called the one of the plurality of overriding methods comprises comparing the call context information to selected content of a call stack.

50. (New) The method of claim 49, wherein the target call site is determined to have called the one of the plurality of overriding methods if stored call context information matches the selected content of the call stack.

51. (New) The method of claim 50, wherein the selected content of the call stack is a call to the one of the plurality of overriding methods, and wherein the target call site is determined to have called the one of the plurality of overriding methods if the stored call context information matches the selected content of the call stack.

52. (New) A computer-implemented method for debugging code, comprising:

(a) while execution is halted at a first point in the code, receiving a user selection of a target call site of the code, the user selection being made through a graphical user interface, and the target call site comprising a call to any one of a plurality of overriding methods;

(b) in response to receiving the user selection, setting a run-into breakpoint at each respective entry point of the plurality of overriding methods;

(c) determining call context information identifying a location of the target call site in the code; and

(d) for each of the run-into breakpoints encountered during execution of the code:

determining whether the target call site called one of the plurality of overriding methods, based on the call context information; and

if so:

halting execution of the code at the target call site.

53. (New) The method of claim 52, further comprising:

repeating each of the steps (a)-(d) for a plurality of target call sites, wherein each target call site has an associated method of an associated plurality of overriding methods, an associated breakpoint at each respective associated entry point of the associated plurality of overriding methods, and associated call context information; and

automatically removing at least one of the associated breakpoints upon determining that one of the plurality of target call sites has called the associated method based on the associated call context information.

54. (New) The method of claim 52, wherein determining whether the target call site called the one of the plurality of overriding methods comprises comparing the call context information to selected content of a call stack.

55. (New) The method of claim 54, wherein the target call site is determined to have called the one of the plurality of overriding methods if stored call context information matches the selected content of the call stack.

56. (New) The method of claim 54, wherein the selected content of the call stack is a call to the one of the plurality of overriding methods, and wherein the target call site is determined to have called the one of the plurality of methods if the stored call context information matches the selected content of the call stack.

57. (New) The method of claim 52, wherein the code id object-oriented, and the method further comprises identifying the plurality of overriding methods.

58. (New) The method of claim 57, further comprising, prior to identifying the plurality of overriding methods, determining that the method is associated with an object.

59. (New) The method of claim 57, wherein identifying the plurality of overriding methods comprises traversing a class hierarchy, and locating each matching member method according to the selected target call site.

60. (New) A computer readable storage medium containing a program which, when executed, performs an operation for debugging code, comprising:

- while execution is halted at a first point in the code, receiving a user selection of a target call site of the code, the user selection being made through a graphical user interface, and the target call site comprising a call to any one of a plurality of overriding methods;

- in response to receiving the user selection, setting a run-into breakpoint at each respective entry point of the plurality of overriding methods;

- determining call context information identifying a location of the target call site in the code;

- encountering one of the run-into breakpoints;

- determining whether the target call site called one of the plurality of overriding methods, based on the call context information; and

- halting execution of the code at the target call site.

61. (New) The computer readable storage medium of claim 60 wherein the call context information unambiguously identifies the code.

62. (New) The computer readable storage medium of claim 60, wherein determining whether the target call site called the one of the plurality of overriding methods comprises comparing the call context information to selected content of a call stack.

63. (New) The computer readable storage medium of claim 62, wherein the target call site is determined to have called the one of the plurality of overriding methods if stored call context information matches the selected content of the call stack.

64. (New) The computer readable storage medium of claim 63, wherein the selected content of the call stack is a call to the one of the plurality of overriding methods, and wherein the target call site is determined to have called the one of the plurality of overriding methods if the stored call context information matches the selected content of the call stack.

65. (New) A computer readable storage medium containing a program which, when executed, performs an operation for debugging code, comprising:

- (a) while execution is halted at a first point in the code, receiving a user selection of a target call site of the code, the user selection being made through a graphical user interface, and the target call site comprising a call to any one of a plurality of overriding methods;

- (b) in response to receiving the user selection, setting a run-into breakpoint at each respective entry point of the plurality of overriding methods;

- (c) determining call context information identifying a location of the target call site in the code; and

- (d) for each of the run-into breakpoints encountered during execution of the code:

- determining whether the target call site called one of the plurality of overriding methods, based on the call context information; and

- if so:

- halting execution of the code at the target call site.

66. (New) The computer readable storage medium of claim 65, further comprising:
repeating each of the steps (a)-(d) for a plurality of target call sites, wherein each target call site has an associated method of an associated plurality of overriding

methods, an associated breakpoint at each respective associated entry point of the associated plurality of methods, and associated call context information; and

automatically removing at least one of the associated breakpoints upon determining that one of the plurality of target call sites has called the associated method based on the associated call context information.

67. (New) The computer readable storage medium of claim 65, wherein determining whether the target call site called the one of the plurality of overriding methods comprises comparing the call context information to selected content of a call stack.

68. (New) The computer readable storage medium of claim 67, wherein the target call site is determined to have called the one of the plurality of overriding methods if stored call context information matches the selected content of the call stack.

69. (New) The computer readable storage medium of claim 67, wherein the selected content of the call stack is a call to the one of the plurality of overriding methods, and wherein the target call site is determined to have called the one of the plurality of overriding methods if the stored call context information matches the selected content of the call stack.

70. (New) The computer readable storage medium of claim 65, wherein the code is object-oriented, and the method further comprises identifying the plurality of overriding methods.

71. (New) The computer readable storage medium of claim 70, further comprising, prior to identifying the plurality of overriding methods, determining that the method is associated with an object.

72. (New) The computer readable storage medium of claim 70, wherein identifying the plurality of overriding methods comprises traversing a class hierarchy, and locating each matching member method according to the selected target call site.

73. (New) A computer, comprising:

a memory;
a processor;

code under debug resident in the memory, the code comprising at least one target call site comprising a call to any one of a plurality of overriding methods, the target call site being selected by the user through a graphical user interface while execution is halted at a first point in the code;

a breakpoint data structure resident in the memory and configured for storing at least context information indicating a location of the target call site within the code; and

a debugger program resident in the memory and which, when executed by the processor, is configured to interrupt execution of the code under debug in response to encountering a breakpoint set on one of the plurality of overriding methods if the selected target call site called the one of the plurality of overriding methods with reference to the context information.

74. (New) The system of claim 73, further comprising a caller data structure resident in the memory and configured for storing at least callers of overriding methods in the code as encountered during an execution path; and wherein the debugger program is configured to determine whether the routine is called from the target call site by comparing the context information to a caller stored in the caller data structure.

75. (New) The system of claim 73, wherein the code under debug is object-oriented.

76. (New) The system of claim 73, wherein the debugger program is further configured to place a run-into breakpoint at each respective entry point of the plurality of overriding methods.